# **SRS Document**

## **Team ID : 45 - Talabat**

## **Super Visor TA: Radwa Mostafa**

نادين مصطفى علي أحمد

20201700902

sec 8

نانسي ابراهيم عطية ابراهيم

20201700906

sec 9

نوران وليد محمد أحمد المرسي

20201700936

sec 9

عبدالرحمن محمد نجيب محي الدين محمود

20201700449

sec 4

عبدالله يحيى سليمان يحيى

20201700466

sec 4

عبدالعزيز حسام عبدالعزيز عبدالفتاح

20201700457

Sec 4

## 1. Introduction:

The main goal of this software system is to create an efficient online food ordering and delivery platform for customers and restaurants which has become increasingly necessary due to the high demand for such services, particularly in urban areas where people are short on time.

The system includes two types of users, an admin and customer. Customer can create an account and login, while the admin already has an account and only logs in.

The customer selects a restaurant, orders food, also has the ability to benefit from promocodes to have discounts, and can also view their history of orders.

The admin's role is to manage promocodes and manage restaurants.

The system's features include the ability for customers to view restaurants. Additionally, the system is compatible with other tools, such as payment gateways and facilitates online transactions.

By fulfilling those objectives, the system overall should achieve satisfaction for customers, by good user experience, available application all the time, fast, stable, and durable with large number of users, and one of the most important things also is that user's data must be secure.

## 2. User Requirements:

(1) The user should create an account, and login. Also, the admin must be able to login.
(2) Customer should be able to select restaurant from the list of restaurants.
(3) Customers order food and can add promocodes to their order, to obtain useful discounts on their orders.
(4) Customers should view past orders they made previously.
(5) Admin manages promocodes and restaurants.
(6) The system should be responsive and accessible.
(7) The system should be available all the time, handle many orders and be durable with pressure.
(8) System should be secure to protect user's data.
(9) System should be available on different devices with responsive design that is comfortable for the user.

## 3. Functional Requirements:

(1)

Functional name: Login

Description: The user should enter their username and password to use the system

Inputs: Username, password

Source: User and Admin

Pre-condition: Open the application

Post- condition: Other functions

Output: Login successfully or failed to login

(2)

Functional name: Create Account

Description: If the user doesn’t have an account, they should create one

Inputs: Username, password, phone number, address

Source: User

Pre-condition: Open the application

Post- condition: Login

Output: Account created successfully

(3)

Functional name: Select Restaurant

Description: Allow customers to choose a restaurant from the list of available restaurants by name.

Inputs: none

Source: User

Pre-condition: Customer is logged in

Post- condition: Ordering Food

Output: list of restaurants including different cuisines

(4)

Functional name: Ordering Food

Description: allow customer to order from a selected restaurant, where he adds the item, quantity, and delivery address

Inputs: item, quantity, and delivery address

Source: User

Pre-condition: customer selected the restaurant

Post- condition: order confirmed page

Output: order confirmation (Message Box)

(5)

Functional name: Use Promocode

Description: the user enters promocode in text box to get discount for their orders

Inputs: text box (promocode)

Source: user

Pre-condition: Ordering Food

Post- condition: show price

Output: discount applied or not (activated, used, expired)

(6)

Functional name: View Orders History

Description: the user views all past orders he did

Inputs: none

Source: user

Pre-condition: Ordering Food

Post- condition: back to home page

Output: display all the past orders

(7)

Functional name: Manage Restaurants

Description: The admin can add, edit, or delete the restaurants

Inputs: if add or edit is chosen, the admin enters new restaurants

Source: admin

Pre-condition: Admin should be logged in

Post- condition:  list of restaurants

Output: display changed restaurants

(8)

Functional name: Manage Promocodes

Description: The admin can add, or delete the promocodes

Inputs: if add is chosen, the admin enters new promocodes

Source: admin

Pre-condition: Admin should be logged in

Post- condition: promocodes

Output: display changed promocodes

## 4. Non-Functional:

(1) Security: The system must be secure and protect user's data for example it can be encrypted

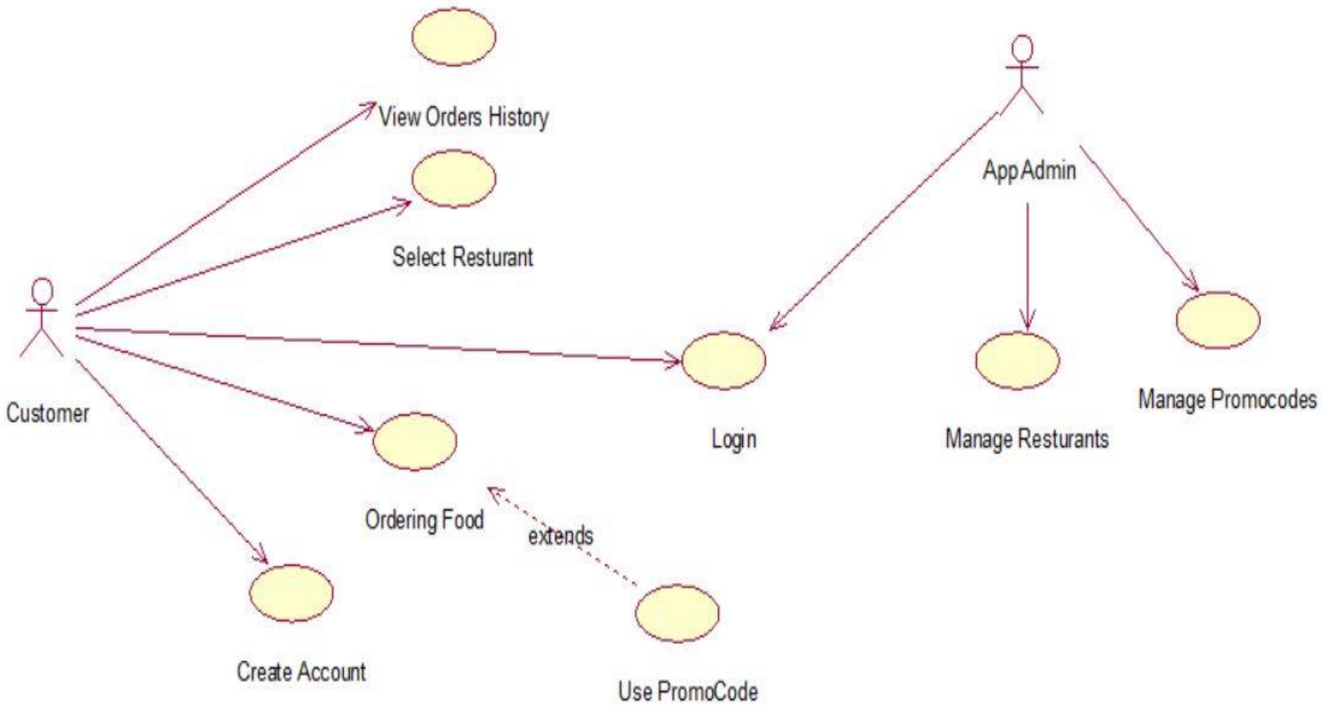
(2) Availability: The system should be available and accessible to user all the time

(3) Performance: The system should be able to handle large number of users with their orders at the same time, with fast response time and high performance

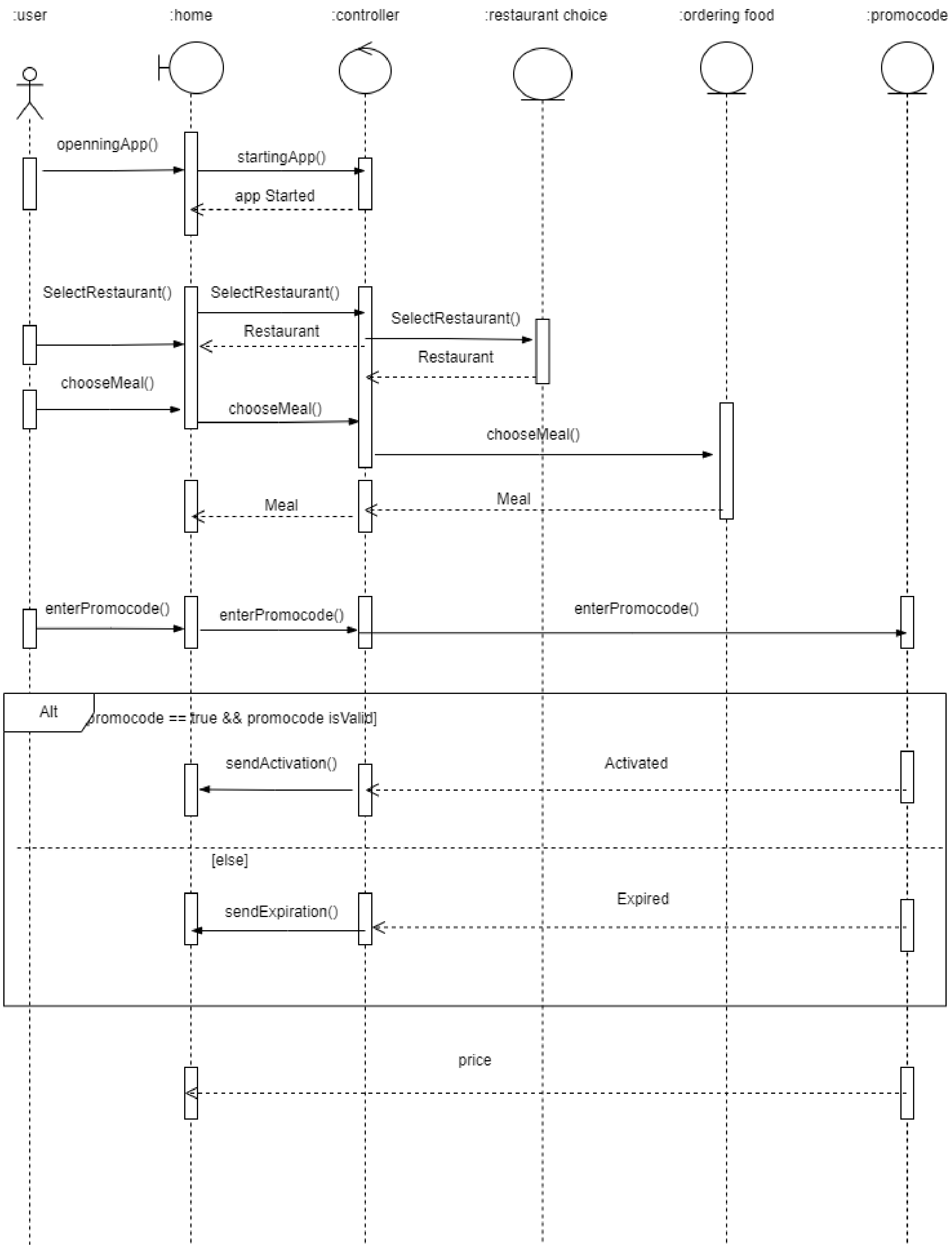
(4) Usability: The system should have a simple easy interface for the user to interact with, in addition to a good experience in ordering and using the system. Also, the system should have a responsive design on all devices.

(5) Scalability: The system should be able to handle the growth of the number of users without crashing or performing slowly and should be able to scale up and down quickly, as needed.

- **Use Case Diagram:**



- **Sequence Diagram:**


:user
:home
:controller
:restaurant choice
:ordering food
:promocode
openningApp()
startingApp()
app Started
SelectRestaurant()
SelectRestaurant()
SelectRestaurant()
Restaurant
Restaurant
chooseMeal()
chooseMeal()
chooseMeal()
Meal
Meal
enterPromocode()
enterPromocode()
enterPromocode()
Alt
promocode == true && promocode isValid]
sendActivation()
Activated
[else]
sendExpiration()
Expired
price